LLNL-TR-405048

LAWRENCE LIVERMORE NATIONAL LABORATORY

Use of Multi-attribute Utility Functions in Evaluating Security Systems

Carol Meyers, Alan Lamont, and Alan Sicherman

June 2008

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory in part under Contract W-7405-Eng-48 and in part under Contract DE-AC52-07NA27344.

Use of Multi-attribute Utility Functions in Evaluating Security Systems

Carol Meyers, Alan Lamont, and Alan Sicherman

Lawrence Livermore National Laboratory

Introduction

In analyzing security systems, we are concerned with protecting a building or facility from an attack by an adversary. Typically, we address the possibility that an adversary could enter a building and cause damage resulting in an immediate loss of life, or at least substantial disruption in the operations of the facility. In response to this setting, we implement security systems including devices, procedures, and facility upgrades designed to a) prevent the adversary from entering, b) detect and neutralize him if he does enter, and c) harden the facility to minimize damage if an attack is carried out successfully. Although we have cast this in terms of physical protection of a building, the same general approach can be applied to non-physical attacks such as cyber attacks on a computer system.

A rigorous analytic process is valuable for quantitatively evaluating an existing system, identifying its weaknesses, and proposing useful upgrades. As such, in this paper we describe an approach to assess the degree of *overall* protection provided by security measures. Our approach evaluates the effectiveness of the individual components of the system, describes how the components work together, and finally assesses the degree of overall protection achieved. This model can then be used to quantify the amount of protection provided by existing security measures, as well as to address proposed upgrades to the system and help identify a robust and cost effective set of improvements. Within the model, we use *multiattribute utility functions* to perform the overall evaluations of the system.

Background to the Analytic Problem

In evaluating a security system, certain sets of security measures must work together. Generally, different security system components can either *complement* or *compensate* each other. In the complementary case, two or more security measures must work together to provide an effective security function. For example, to prevent an adversary from entering a facility, there must be barriers around the facility (such as fences or walls) that make it difficult to enter the facility except through authorized entry points. In addition, there must also be effective authorization checks at those entry points to ensure that an unauthorized person cannot simply walk in. These measures complement each other and have the form of an AND condition in a fault tree analysis. Other measures compensate for each other and instead have the form of an OR condition; for example, an adversary within the building might be detected by an electronic sensor, OR he might be detected by an alert employee. Combining over both scenarios, this leads to a fault tree structure for the evaluation function: a series of AND and OR conditions that measure the overall possibility of preventing or mitigating the damage caused by an adversary.

The standard fault tree model is a probabilistic model in which the AND and OR conditions are *hard* conditions: a condition completely fails or succeeds depending on whether or not the corresponding sub-conditions fail or succeed. The *multiattribute utility framework* can be used as a *generalization* for a fault tree analysis. It can be calibrated to provide either hard AND and OR conditions or *soft* AND and OR conditions, such that there may be partial success or failure for a set of conditions. The ability to model soft conditions is especially useful when the data are too subtle, complex, or difficult to obtain for a full probabilistic analysis. In its extreme, the multiattribute utility model can reduce to a fault tree, but it is also sufficiently general to avoid the limitations of such analyses.

As an illustration of this situation, suppose two components work together as an AND condition. In a fault tree analysis with hard conditions, the failure of one component would mean the failure of the entire function. With soft conditions, the failure (or absence) of one component might severely degrade, but not eliminate, the overall effectiveness. In a preceding example, it was pointed out that external barriers should be used with authorization checks at the entrance points in order to have effective access control; however, if there were very weak authorization checks, the function would not be entirely impaired. Casual authorization checks coupled with strong external barriers are *considerably* better than no access controls at all. Multiattribute utility theory can capture this preference, while a strict probabilistic method cannot.

In what follows, we begin by outlining the general multiplicative form of a multiattribute utility function. We discuss when such forms are useful and how they are represented algebraically. We also show how multiplicative forms can be used to model both *compensatory* and *complementary* interactions, and how they may be calibrated to represent both hard and soft AND and OR conditions. For each interaction, we discuss the *full* and *weak* archetypal representations that are used in practice, as well as the asymptotic utility behavior associated with each representation. We then introduce the *additive* form of the utility function, which is a special case that is intermediate between the AND and OR cases. We discuss the algebraic representation of such forms and when they may be appropriate in practice.

We next represent the spectrum of multiplicative forms in terms of the range of a particular parameter. We discuss techniques for eliciting such parameters and calibrating utility functions in general. We conclude by addressing renormalization techniques that can be useful in the elicitation of strongly complementary interactions. Throughout the paper, our focus is on calibration techniques of the ‘quick and dirty’ variety, which avoid the strain on time and resources associated with a full utility calibration while retaining much of the rigor and formalism.

This paper is intended as a supplement to a standard treatment of multiattribute utility theory, as can be found in Keeney and Raiffa [1]. The theory and functions in this paper have been developed over years of practical research at Lawrence Livermore National Laboratory and other institutions.

Multiattribute Utility Functions for Security Systems

Utility functions are used to evaluate the desirability of a set of conditions, and to compare the desirability of one set of conditions to another. This can be straightforward when there is a single overall consideration, such as the total cost of a project; however, in other cases the evaluation may involve several issues at once. For instance, we might be concerned with both the cost of a project and the total time to completion. In this case, there is a tradeoff: a decision maker might prefer a somewhat higher cost in order to have a shorter completion time. Multiattribute utility has been developed to provide a formal structure for preferences that can include more than one condition (or attribute) at once.

The core of multiattribute utility theory is the use of a pragmatic *aggregation function* for combining the single-utility functions from each of the system components. The general expression of this aggregation is a *multiplicative form*. Such forms allow for an interaction or synergy between the components under consideration, just as we desire in the evaluation of security systems. We now present the algebraic representation of the multiplicative form, followed by a discussion of how such forms can be used to represent *compensatory* and *complementary* interactions between components. The *additive* form, a special case in which each of the components is treated separately, is discussed later.

Algebraic Representation

In assessing a system, we break the security systems in the facility down into basic components and address the conditions of the individual components. Such components can include items such as electronic sensors, the training and placement of personnel, the ability to respond to alarms, and the strength of barriers. Each component i is given a score, denoted x_i. The present discussion does not focus on how this is achieved (for further information, see Keeney and Raiffa [1]). The score x_i is based on objective, observable conditions (such as how many people are in an area, how frequently sensors are tested and maintained, and how long it would take an adversary to break a lock). This score does not necessarily directly reflect the effectiveness of that component. Each score is then translated into a rating of the component using the corresponding *single attribute utility function*, $U_i(x_i)$. The determination of these single attribute utility functions is part of the overall assessment process.

Using these single attribute functions, the *multi-attribute utility function* is of the form:

$$U(x_1, x_2, ..., x_n) = \frac{\prod_i [1 + Kk_i U_i(x_i)] - 1}{K}.$$

Here,

$U_i(x_i)$ = the single-attribute utility value for attribute i with score x_i (ranges from 0 to 1),

k_i = a parameter from the tradeoff for component i (which we address later), for all i, and

K = a normalization constant, ensuring that the utility values are scaled over the component range space between 0 and 1.

A useful representation of the function is obtained by setting $c_i = Kk_i$ for all i, which leads to the following form:

$$U(x_1, x_2, ..., x_n) = \frac{\prod_i [1 + c_i U_i(x_i)] - 1}{\prod_i [1 + c_i] - 1}.$$

In this, we are also using the fact that the parameter $K = \prod_i [1 + c_i] - 1$, which we obtain by observing that the greatest value the numerator can achieve is exactly equal to $\prod_i [1 + c_i] - 1$. Scaling by this factor of K ensures that the overall utility function is between 0 and 1.

We can illustrate the behavior of the utility function using a simple case of two variables. In this situation, the utility function can be simplified as:

$$U(x_1, x_2) = \frac{(1 + c_1 U_1(x_1))(1 + c_2 U_2(x_2)) - 1}{(1 + c_1)(1 + c_2) - 1}.$$

Using the fact that $c_i = Kk_i$, this can be rewritten as:

$$U(x_1, x_2) = k_1 U_1(x_1) + k_2 U_2(x_2) + K k_1 k_2 U_1(x_1) U_2(x_2).$$

We can address some of the basic characteristics of the utility function by examining this last equation. The first two terms of the expression provide a linear interaction between the overall utility and the single-attribute utility functions. The last term is a multiplicative *interaction* term. The settings of the k_is and K determine how these linear and multiplicative terms interact. In general, the value of K can be negative, positive, or approach 0 (a singularity in the multiplicative equation occurs if K is exactly equal to 0). In addition, the sum of the k_is can be less than, equal to, or greater than 1. The values of K and the k_is are not independent. In the case where each of the component utilities is at their maximum value of 1, the overall utility is 1, giving the relation:

$$1 = k_1 + k_2 + K k_1 k_2.$$

From this relation, and the fact that the k_is are positive, we can deduce that if K equals 0, the sum of the k_is = 1; if K is negative, the sum of the k_is > 1; and if K is positive, the sum of the k_is < 1.

Varieties of Interactions

As the value of K ranges from negative, to 0, to positive, the overall utility function can reflect three different types of interactions between individual components. We outline each of these.

- In the *compensatory* case, performance of one component can make up for the lack of performance by other components. In the extreme, the decision maker might think, "If just one of these components is at its best level, then I'm set."
- In the *complementary* case, a good performance by one component is less important than balanced performance across the components. In the extreme, the decision maker might think, "If just one of these components is at its worst level, then the whole system is kind of bad."
- In the *additive* case the performance of one component does not interact with the value of the other components.

In what follows, we illustrate cases where the components all have *equal c_i values*. This assumption is not true in general, but can be reasonable in many applications since attribute ranges can often be scaled to achieve similar weights (see [1] for details).

Compensatory Case

We now discuss the structure of compensatory interactions, both qualitatively and algebraically. In the two-component case, a *compensatory* relationship means that a high utility on one component can partially compensate for a low utility on the other. Figure 1 illustrates a strongly compensatory case. If we examine the upper left corner of the graph, where x_1=0 and x_2=1, we see that the utility is slightly greater than 0.9, in spite of the fact that x_1 is at its lowest level. Thus, the fact that x_2 is at a high level *almost completely compensates* for the fact that x_1 is at its lowest level. We also note that because the iso-utility curves are concave, the overall utility improves slowly as x_1 is improved from 0.

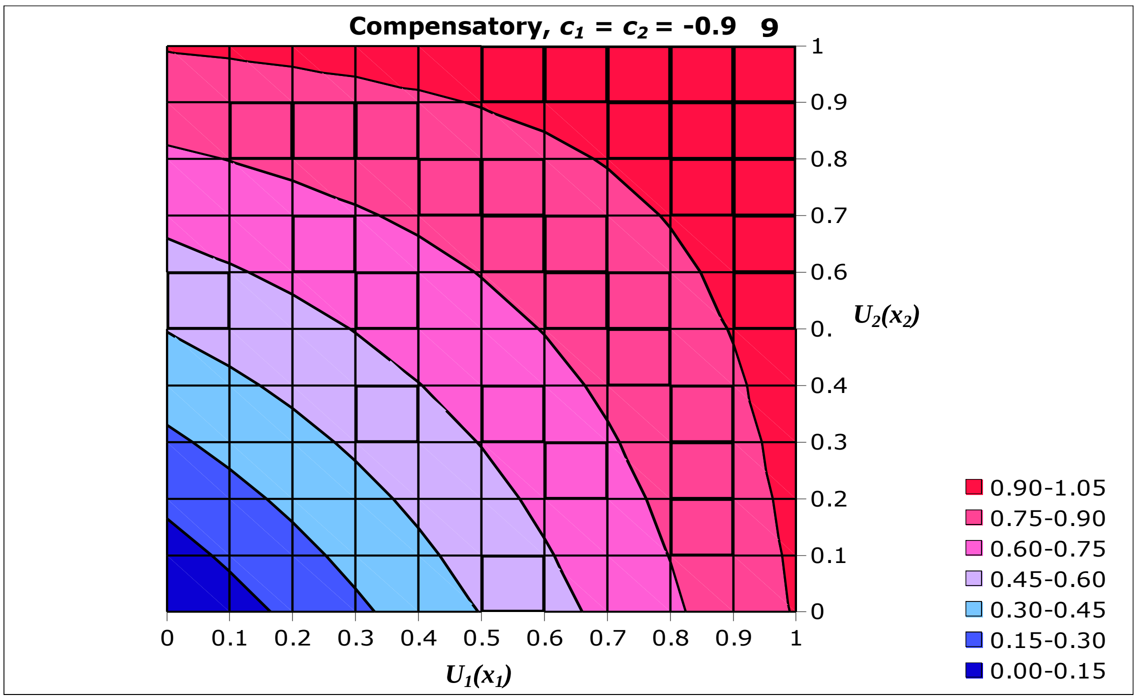

Figure 1: Iso-utility curves for compensatory case. The parameters c_1 and c_2 have been set equal to a value of -0.9, which makes this a strongly compensatory case.

Strong Compensatory Case

The *strong compensatory case* can be thought of as a s*trong OR*, where the overall utility evaluates to 1 if *any* of the components' utility functions evaluate to 1.

Algebraically, this interaction is obtained when $\boldsymbol{c_i = -1}$ for all components *i*. This corresponds to a utility function of the type:

$$U(x_1, x_2, \ldots, x_n) = 1 - \prod_i [1 - U_i(x_i)].$$

Note that if *any* of the utility functions $U_i(x_i) = 1$, then the entire utility function evaluates to 1. This implies that a single component at its best level causes the entire utility function to be at its best level.

Weak Compensatory Case

In many applications, the assumptions of the full compensatory case are too restrictive. The *weak compensatory case* represents a more moderate version of the compensatory case. In this case, the best performance of a single component *partially* compensates for poor performance by the other components. This can be thought of as a *weak OR*, where the overall utility achieves at least a certain *intermediate* value if any of the single-variable utility functions evaluate to 1.

Algebraically, such an interaction is obtained when $\mathbf{-1 < c_i < 0}$ for all i. This corresponds to utility functions of the type:

$$U(x_1, x_2, \ldots, x_n) = \frac{1 - \prod_i [1 + c_i U_i(x_i)]}{1 - (1 + c_i)^n}.$$

The *Archetypal Weak Compensatory case* is obtained when $\mathbf{c_i = -.5}$ for all components i. Asymptotically, if *one* component's utility function evaluates to 1 and *all other* components' utility functions evaluate to zero, the overall utility is equal to .5 as the number of components goes to infinity. This is less extreme than the full compensatory case, where the overall utility would be equal to 1. Algebraically:

$$\begin{aligned} U(x_1, x_2 \ldots, x_n) &= \frac{1 - (1 - .5)(1)^{n-1}}{1 - (1 - .5)^n} \\ &= .5 \qquad \text{as } n \to \infty. \end{aligned}$$

This archetypal case can be appropriate in situations where there is a compensatory interaction between the components, but the strong compensatory case is deemed too extreme.

Other weak compensatory variants can be obtained by modifying the value of c_i that is chosen. For the values $-1 < c_i < -.5$, we can obtain a '*stronger*' compensatory interaction. Similarly, for the values $-.5 < c_i < 0$, we can obtain a '*weaker*' compensatory interaction. Which variety is appropriate for the problem in consideration is determined via elicitation and discourse with the decision-maker.

In general, given a value of c_i between -1 and 0, the asymptotic behavior of a weak compensatory utility function on the solution (1,0,…0) tends to:

$$U(x_1, x_2 \ldots, x_n) = \frac{1-(1+c_i)}{1-(1+c_i)^n}$$

$$= -c_i \qquad \text{as } n \to \infty.$$

This formula can be used to choose other values of c_i that result in ‘stronger’ and ‘weaker’ compensatory interactions, as appropriate. Graphically, the choice of c_i affects the minimum utility that can be obtained in this case as follows:

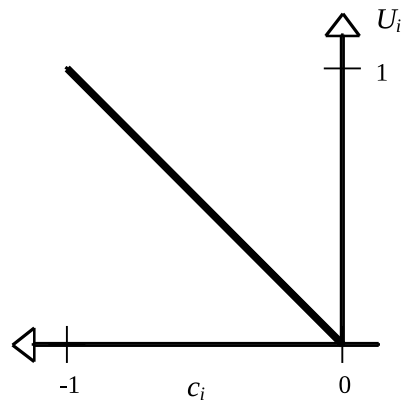

Figure 2: Minimum Utility Guaranteed as a Function of c_i Value.

This graph can also help analyze the sensitivity of the observed results and how they depend on the chosen c_i value.

Complementary Case

Two components have a *complementary* relationship when they reinforce each other, or when both are needed to perform a function.

Figure 3 illustrates a strong complementary interaction. Examining the upper left corner at x_1=0 and x_2=1, we see the utility is quite low at about 0.14, even though one of the components is at full value. In complementary cases such as this, the iso-utility curves are convex. Consequently, as x_1 is improved from 0, the utility improves rapidly.

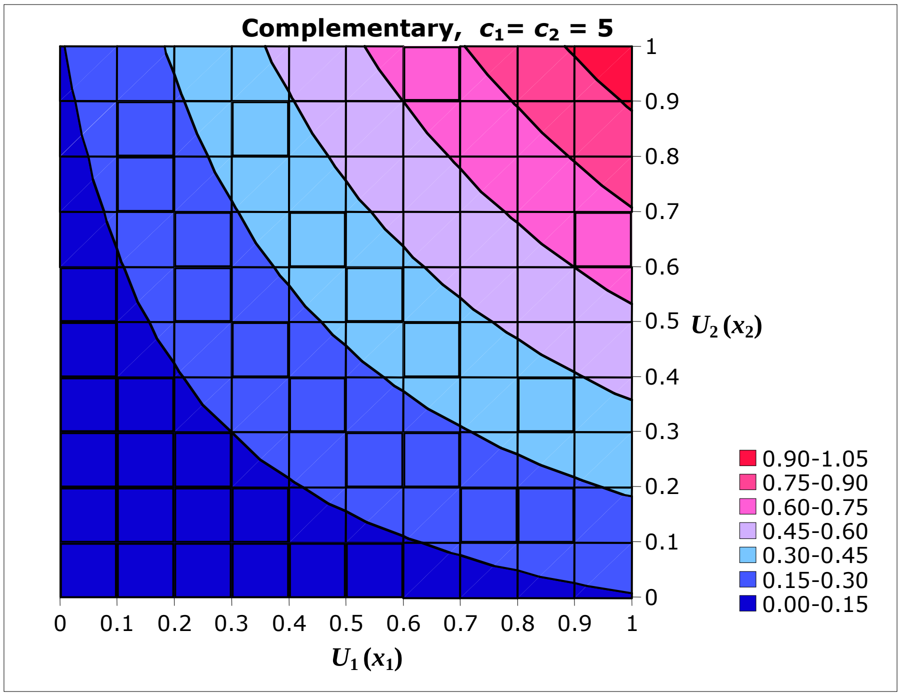

Figure 3: Iso-utility curves for complementary case. The parameters $c_1$and c_2 have been set equal to a value of 5, which makes this a strongly complementary case.

Analogous to the compensatory case, there are two main varieties of complementary interactions: the *strong complementary case* and the *weak complementary case*.

Strong Complementary Case

In a *strong complementary case*, the worst performance by one component entirely cancels out the performance of the other components. This can be thought of as a *strong AND*, where the overall utility evaluates to 0 if *any* of the components' utility functions evaluate to 0. Algebraically, this kind of interaction is obtained when $\boldsymbol{c_i} = \infty$ for all components i. This corresponds to a utility function of the type:

$$U(x_1, x_2,\ldots, x_n) \approx U_1(x_1)U_2(x_2)..U_n(x_n).$$

Note that if *any* component utility function $U_i(x_i) = 0$, then the entire utility function evaluates to 0. This implies that a single component at its worst level causes the entire utility function to be at its worst level. Thus the performance of a single component is less important than *balanced* performance across different components.

Weak Complementary Case

Occasionally the assumptions of the full complementary case can be too extreme. In certain situations, as described in the introduction, it is desirable to have at least a *partial* sense of progress as individual component utility values are improved. For such situations, the *weak complementary case* represents a more moderate version of the complementary case. In this instance, a single component at its worst level *partially* cancels out the performance of the other components. This can be thought of as a *weak AND*, where the overall utility achieves at most a certain *intermediate* value if any of the components' utility functions evaluate to 0.

Algebraically, such an interaction is obtained when $\mathbf{0 < c_i < \infty}$ for all i. This corresponds to utility functions of the type:

$$U(x_1, x_2, \ldots, x_n) = \frac{\prod_i [1 + c_i U_i(x_i)] - 1}{(1 + c_i)^n - 1}.$$

The *Archetypal Weak Complementary case* is obtained when $\mathbf{c_i = 1}$ for all components i. Asymptotically, if *one* component's utility function evaluates to 0 and *all other* components' utility functions evaluate to 1, the overall utility is equal to .5 as the number of components goes to infinity. This is less extreme than the full complementary case, where the overall utility would be equal to 0. Algebraically:

$$\begin{aligned} U(x_1, x_2 \ldots, x_n) &= \frac{(2)^{n-1} - 1}{(2)^n - 1} \\ &= .5 \quad \text{as } n \to \infty. \end{aligned}$$

This archetypal case is used when there is a complementary interaction between the components, but the strong complementary case is deemed too severe.

Similar to the compensatory case, weak complementary variants can be obtained by modifying the value of c_i that is chosen. For the values $0 < c_i < 1$, we can obtain a '*weaker*' complementary interaction. Similarly, for the values $1 < c_i < \infty$, we can obtain a '*stronger*' complementary interaction. Which variety is appropriate is determined via elicitation with the decision-maker.

In general, given a value of c_i between 0 and ∞, the asymptotic behavior of a weak complementary utility function on the solution (0,1,…,1) tends to:

$$\begin{aligned} U(x_1, x_2 \ldots, x_n) &= \frac{(1 + c_i)^{n-1} - 1}{(1 + c_i)^n - 1} \\ &= \frac{1}{1 + c_i} \quad \text{as } n \to \infty. \end{aligned}$$

This formula can be used to choose other values of c_i that result in 'stronger' and 'weaker' complementary interactions, as appropriate. Graphically, the choice of c_i affects the maximum utility that can be obtained as demonstrated in Figure 4. Again this graph can be used to help analyze the sensitivity of the results and dependence on the c_i value.

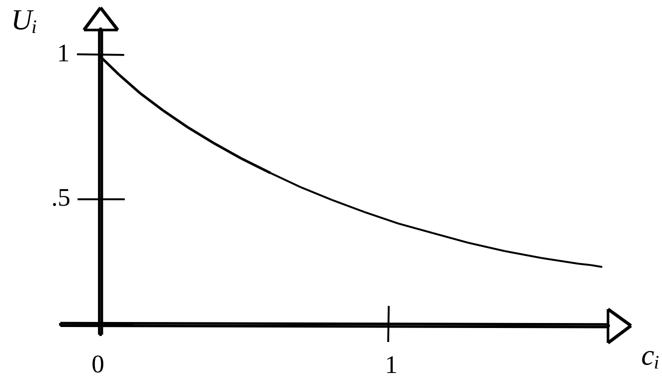

Figure 4: Maximum Utility Obtainable as a Function of c_i Value.

Additive Case

The additive case is a special case where there is *no interaction* between the components. Here, the total utility is simply the weighted sum of the utilities of the individual components. Figure 5 shows the iso-utility curves for an additive case. In this example, the k_is are equal and sum to 1. Examining the upper left corner, at $x_1 = 0$ and $x_2 = 1$ we see that the overall utility is 0.5. This reflects that we only get credit for x_2, and there is no penalty for the fact that $x_1 = 0$. Note that the utility is 0.5 in this case because the k_is are equal. More generally, in the additive case the utility of the corner will depend on the ratio of the k_is.

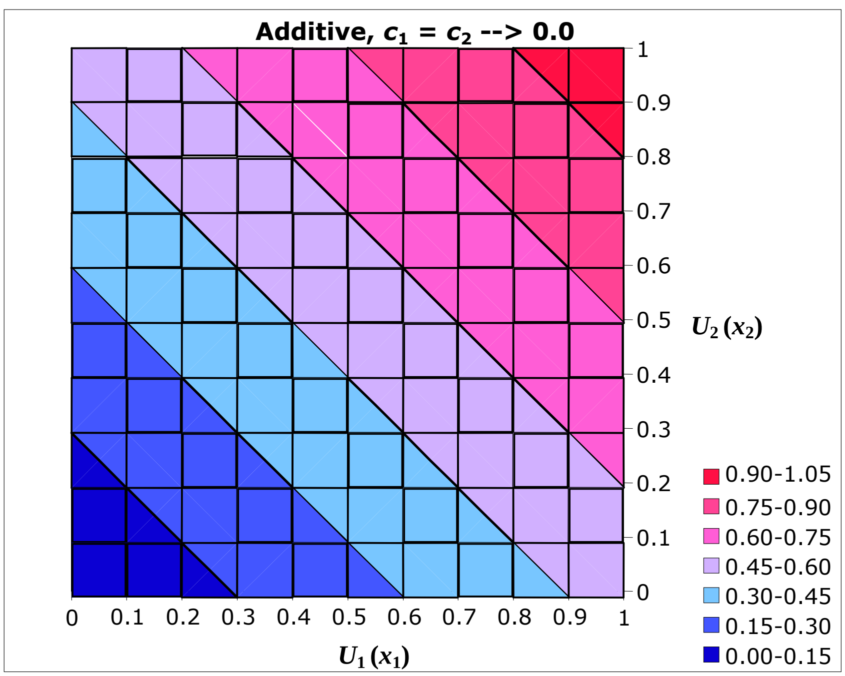

Figure 5: Additive case. The parameters c_1 and c_2 approach a value of 0.0.

Additive forms are appropriate for systems in which the components to be evaluated exhibit *little interaction* with each other. Heuristically, the overall utility of a system can be expressed as the sum of its parts. If the utility of one such component evaluates to zero, then the full utility value cannot be achieved, but at the same time it does not diminish the contributions of the other components. The additive form is also used in situations where the ranges of component performance (best level to worst level) are not too broad or extreme. In such cases, the limitations of the additive form are not as pronounced as when components can evaluate to greatly different levels, and the simpler additive form may be preferred.

The additive form is a special case of the general multi-attribute utility function. The general function *approaches* the additive form as the value of the c_is (and hence, also the value of K) approach 0. The basic utility function for an additive form is as follows:

$$U(x_1, x_2, \ldots, x_n) = k_1 U_1(x_1) + k_2 U_2(x_2) + \ldots + k_n U_n(x_n)$$

where $k_1, k_2, \ldots, k_n$ are nonnegative constants such that $k_1 + k_2 + \ldots + k_n = 1$.

This form is known as *additive* because the k_i terms represent a relative weighting of the various components, and the overall utility is obtained by taking a weighted sum of the individual utility functions.

Summary of Cases

In the previous sections, we observed how the values of c_i, k_i, and K chosen for a multiplicative form can influence the behavior of the form both qualitatively and algebraically. The following table summarizes the relationships between these three values, and what kind of interaction each combination represents.

Value of K	**Sum of the k_is**	**Value of c_is**	**Type of Interaction**
Negative	>1.0	Negative	compensatory
Approaches zero	1.0	Approaches zero	additive
Positive	<1.0	Positive	complementary

Table 1: Relationships between the values of c_i, k_i, and K, and the type of interaction represented.

This table can be used to understand the interplay between these three quantities and how such algebraic parameters can be used to represent different relationships between components of a system.

Further illustrating this phenomenon, Figure 6 addresses the ranges of possible c_i values and how each of these translates into compensatory, complementary, or additive cases. The full and weak versions of each case are detailed, as well as the archetypal representative of each case.

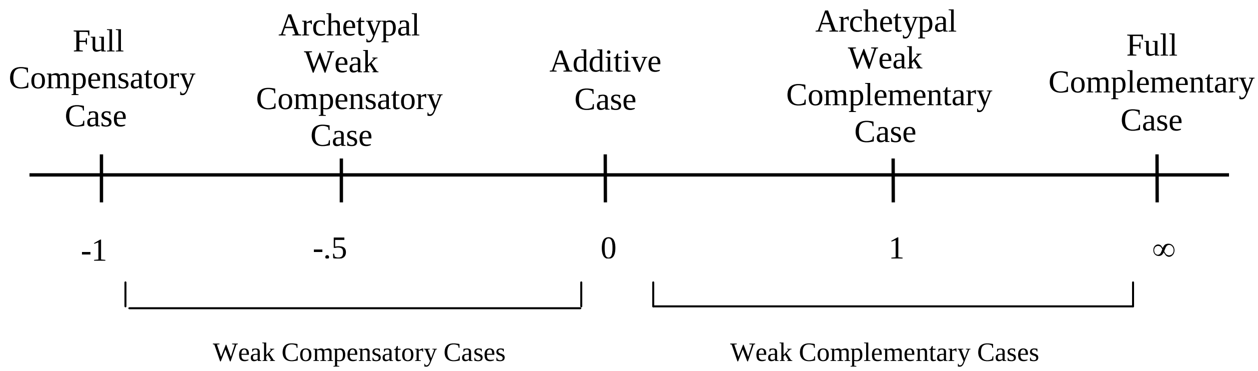

Figure 6: Spectrum of values for c_i and the resulting interactions between components.

Note that as the value of c_i approaches zero, the interaction terms represent less weight in the utility function. Hence in the limit, the multiplicative form approaches an additive form.

Figure 7 shows the impact of the c_i values on the utility value at the corner point of the utility function where $x_1 = 0$ and x_2=1. When the c_is approach -1, the value at the corner point approaches 1. In this case, the fact that x_2 is at its highest level completely compensates for the fact that x_1 is at its lowest level. This case corresponds to the hard OR in a fault tree analysis. At the other extreme, as the c_is go to infinity, the utility at the corner point approaches 0 (the graph is truncated here at $c_i = 3.5$). This corresponds to the hard AND where both components must perform well to achieve functionality. As the c_is approach 0, the function becomes additive and the utility value of the corner point goes to 0.5, indicating the two attributes have no interaction.

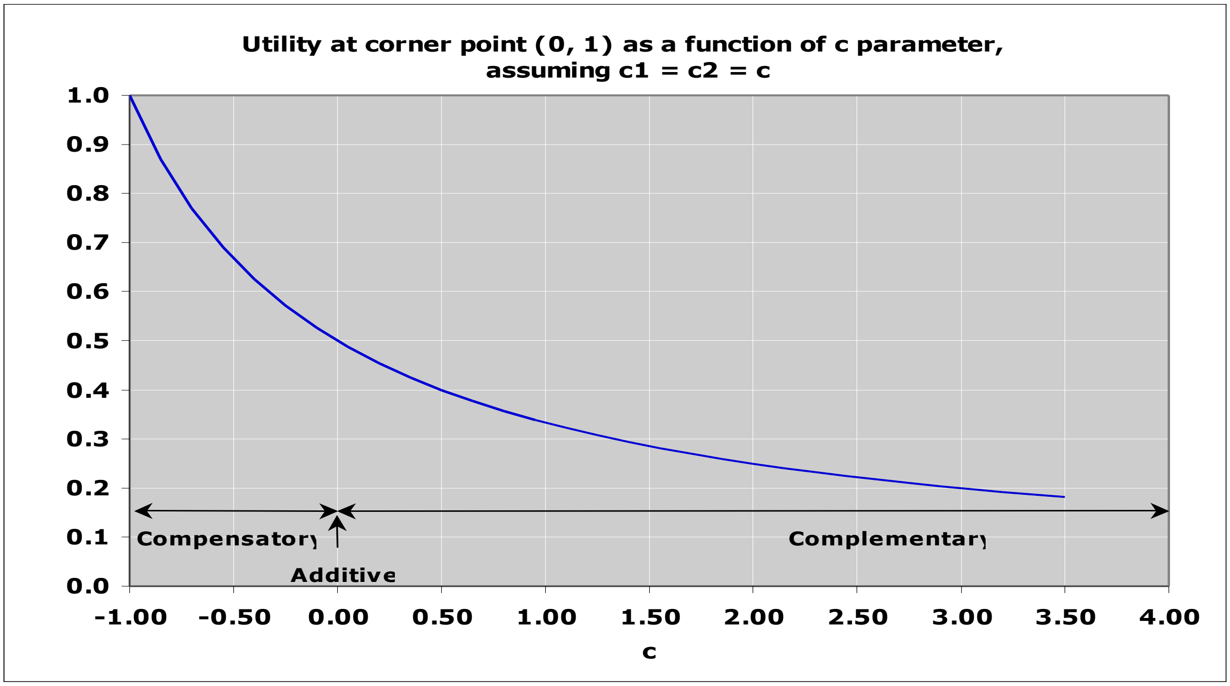

Figure 7: Utility at the corner point (0,1) as a function of the c parameter, assuming the c_i's are equal.

Calibrating the Function

We now address approaches for eliciting the values of c_i for the forms that we have discussed. Our goal is to provide both intuition for how the forms are structured and an exposition of a simple case.

The calibration procedure consists of two main components: first, we determine the *type of interaction* (complementary, compensatory, or additive) evidenced by the attributes under question, and next, we assess the *strength* (strong or weak) of that interaction. In what follows, we assume for the sake of exposition (as in the rest of the document) that all attributes are equally weighted. As before, this assumption is usually reasonable in practice, because attribute ranges can often be scaled to achieve similar weights.

Determining the Type of Interaction

One way to determine the kind of interaction between two attributes is as follows. Suppose that $\{x_1, x_2\}$ represents the state of the attributes in a given situation, and $(U_1(x_1), U_2(x_2))$ represents their corresponding utilities. We then consider tradeoffs of the form in Figure 8:

Figure 8: Lotteries used to determine the kind of interaction between attributes.

In Lottery 1, there is a 50% chance of attribute 1 being at its highest level and attribute 2 at its lowest, and a 50% chance of attribute 1 being at its lowest level and attribute 2 at its highest. In Lottery 2, there is a 50% chance of both attributes being at their highest levels, and a 50% chance of both being at their lowest levels.

If the decision maker prefers Lottery 1 to Lottery 2, then we infer that that the interaction is *compensatory*. Here, having one attribute at its best level can make up for a low level on the other attribute. Conversely, if the decision maker prefers Lottery 2 to Lottery 1, we conclude the interaction is *complementary*. This is because having either attribute at its lowest level is nearly as painful as having both attributes at their lowest levels. Finally, if the decision maker views the lotteries as equally preferable, we say the attributes are *additive*. In this situation, there is little interaction between the attributes and both alternatives are equally appealing.

Determining the Strength of the Interaction

We now address how to determine the strength of the interaction, for attributes exhibiting complementary or compensatory relationships. (For attributes in the additive case, this factor does not apply.)

To assess the strength of a *compensatory* relationship, the decision maker should compare the solution (1, 0) to the solution (1, 1). If both of these alternatives are nearly equally preferable, then the attributes exhibit a *strong compensatory* relationship. Thus, a strong compensatory form (c_i approaches -1) should be used. If instead (1, 1) is preferred to (1, 0) (which in turn is preferred to (.5, .5), as implied by the tradeoff in the previous section), then the attributes display a *weak compensatory* relationship. For most purposes, it is then sufficient to use the *archetypal weak compensatory form* (c_i = -.5). (If a ‘stronger’ or ‘weaker’ weak compensatory form is desired, equations of the type found at the end of the section on compensatory forms can help determine an appropriate value for the c_i parameters.)

To determine the strength of a *complementary* relationship, the solution (1, 0) should be compared to the solution (0, 0). If both of these alternatives are preferred equally by the decision maker, then the attributes exhibit a *strong complementary* relationship. Hence a strong complementary form (c_i approaches ∞, although a value of, say, 5 or greater does represent a strongly complementary relationship) should be used. If (1, 0) is preferred to (0, 0) (which are both preferred less than (.5, .5), as implied by the tradeoff in the previous section), then the attributes have a *weak complementary* relationship. In most situations, we may then use the *archetypal weak complementary form* (c_i = 1). (If a ‘stronger’ or ‘weaker’ weak complementary form is desired, equations such as those found at the end of the section on complementary forms can help determine an appropriate value for the c_i parameters.)

Determining the Ratio of the c_is When They Are Not Equal

Compensatory Case

When the c_is are not equal, without loss of generality we can assume that the solution (1, 0) is preferred to the solution (0, 1). If we can determine that the solution (u_1, 0) is equally preferred to (0, 1), then we can set the ratio as:

$$\frac{c_2}{c_1} = u_1.$$

To determine the values of the c_is, we start by assigning the c_i term with the largest absolute value in the group of attributes being aggregated to the archetypal value (e.g., -.5 for the weak case). We then use the ratios to determine the values of the other c_i terms.

Normalization Issues

Extremely complementary cases can occasionally be difficult to elicit, because they require the decision maker to perform assessments where one component is always at its worst level. Often times decision makers can be uncomfortable relating to components at their worst levels, and as such they may find it hard to make meaningful comparisons.

A method of dealing with this situation is to *renormalize* the c_i values, in such a way that all complementary cases can be assessed using components at their best levels. We briefly describe one such renormalization that has the utilities going from -1 to 0 instead of 0 to 1: that is, $u_i' = u_i - 1$.

In the renormalization, c_i values are converted into a new parameter c_i' as follows:

$$c_i' = \frac{c_i}{1 + c_i}$$

In this new c_i' universe, the ranges for compensatory and complementary interactions have changed. Specifically,

- *Compensatory interactions* correspond to a range of $\mathbf{-\infty < c_i' < 0}$, and
- *Complementary interactions* correspond to a range of $\mathbf{0 < c_i' \leq 1}$.

Now to obtain an appropriate cancellation of terms, one component in the compensatory case must always be at its *best* level. This can make elicitations a lot easier to perform

The renormalization also alters the ranges of full and weak cases as follows. (Note that in practice, the c_i' term is never set exactly equal to 1 in the full complementary case, as it causes a singularity in the transformation between c_i and c_i' values. A value of .9999 would suffice.)

- The *Full compensatory case* corresponds to $\boldsymbol{c_i' = -\infty}$.
- *Weak compensatory cases* correspond to $\boldsymbol{-\infty < c_i' < 0}$.
- The *Archetypal weak compensatory case* corresponds to $\boldsymbol{c_i' = -1}$.
- The *Full complementary case* corresponds to $\boldsymbol{c_i' = 1}$.
- *Weak complementary cases* correspond to $\boldsymbol{0 < c_i' < 1}$.
- The *Archetypal weak compensatory case* corresponds to $\boldsymbol{c_i' = .5}$.

Determining the Ratio of the c_i's When They Are Not Equal

Complementary Case

When the c_i's are not equal, without loss of generality we can assume that the solution (0, -1) is preferred to the solution (-1, 0). If we can determine that the solution (u_1', 0) is equally preferred to (0, -1), then we can set the ratio as:

$$\frac{c_2'}{c_1'} = -u_1'.$$

To determine the values of the c_i's, we assign the c_i' term with the largest absolute value in the group of attributes being aggregated to the archetypal value (e.g., .5 for the weak case). We then use the ratios to determine the values of the other c_i' terms. Finally, we can use the relation

$$c_i = \frac{c_i'}{1 - c_i'}$$

to translate the c_i' terms back to c_i terms.

Concluding Remarks

This paper has covered the structure and function of multiattribute utility functions applied to the evaluation of security systems. In particular, we have addressed the *compensatory*, *complementary,* and *additive* variants of such forms, which as far as we know have never previously been treated with this particular level of technical detail.

We have provided a picture of how changing parameter values affect the interpretation of the aggregation being performed, and how decision maker beliefs may be used to identify the best choice of a multiplicative form. We have also addressed how such parameters may be obtained through the elicitation of experts, as well as when renormalizations of the parameter space may aid in certain varieties of elicitations. Our hope is to provide a solid theoretical basis for future practitioners of multiattribute utility theory in the area of security systems evaluation.

References

[1] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preference and Value Tradeoffs*. John Wiley & Sons, New York, NY, 1976.